

AMENDMENTS TO THE CLAIMS

1 1-7. (Canceled)

1 8. (Previously Presented) A method of controlling use by concurrent users of a
2 distributed resource on a network, wherein use of the resource is limited to a specified
3 maximum number of concurrent users, the method comprising the computer-
4 implemented steps of:
5 providing a distributed lock manager process comprising a plurality of local lock
6 manager processes executing on a corresponding plurality of hosts,
7 wherein each of the plurality of local lock manager processes may grant a lock on the
8 same resource at any time when a number of outstanding locks granted on the
9 same resource is less than the specified maximum number of concurrent users;
10 associating a user identification for each user with one host of the plurality of hosts;
11 and
12 responding to a request for the resource associated with a first user having a first user
13 identification associated with a first host of the plurality of hosts by requesting
14 a lock from a first local lock manager process executing on the first host.

1 9. (Original) A method as recited in Claim 8, wherein during said step of
2 associating a user identification with one host, the first user is associated with the first
3 host based on information indicating that the first user signs onto the network most
4 frequently at a terminal node of the network that uses fewer intervening network
5 devices to pass messages to the first host than to any other host of the plurality of
6 hosts.

1 10. (Original) A method as recited in Claim 8, wherein during said step of
2 associating a user identification with one host, the first user is associated with the first
3 host based on information indicating that the first user signs onto the network most
4 frequently at a terminal node of the network geographically closer to the first host
5 than to any other of the plurality of hosts.

- 1 11. (Original) A method as recited in Claim 8, wherein each local lock manager
2 process maintains a lock data structure associated with the resource and the lock data
3 structure includes a local resource maximum field; and further comprising the steps of
4 generating and storing a value in the local resource maximum field maintained by
5 each local lock manager process such that a summation over all the local lock
6 manager process of the value in the local resource maximum field does not exceed the
7 maximum number of concurrent users.

- 1 12. (Original) A method as recited in Claim 11, wherein a first value in the local
2 resource maximum field maintained by the first local lock manager process is based
3 on a number of users associated with the first host.

- 1 13. (Original) A method as recited in Claim 8, wherein a copy of the distributed
2 resource resides on a host of the plurality of hosts.

- 1 14. (Original) A method as recited in Claim 8, wherein a copy of the distributed
2 resource resides on a computing device that is not among the plurality of hosts.

- 1 15. (Original) A method as recited in Claim 11, further comprising:
2 determining whether a number of outstanding locks granted by the first local lock
3 manager process is less than a particular value stored in the local resource
4 maximum field maintained by the first local lock manager process; and
5 if the number of outstanding locks is less than the particular value, then generating
6 and returning a lock object.

- 1 16. (Previously presented) A method of controlling concurrent users of a
2 distributed resource on a network, the resource limited to a maximum number of
3 concurrent users, the method comprising the computer-implemented steps of:
4 receiving a request for the distributed resource from a client process for a user having
5 a user identification;

6 determining a home location associated with the user identification, the home location
7 indicating a unique host among a plurality of hosts that execute a
8 corresponding plurality of local lock manager processes of a distributed lock
9 manager process,
10 wherein each of the plurality of local lock manager processes may grant a lock on the
11 same resource;
12 sending a request for a lock object for the distributed resource to a first local lock
13 manager process of the distributed lock manager process, the request
14 including the home location;
15 receiving the lock object for the distributed resource from a second local lock
16 manager process executing on the unique host, if a number of outstanding
17 locks granted by the second local lock manager process is less than a value of
18 a local resource maximum defined for the second local lock manager process;
19 and
20 providing access to the resource to the first client only in response to receiving the
21 lock object.

- 1 17. (Original) A method as recited in Claim 16, wherein a first host on which the first
2 local lock manager process executes is different from the unique host on which the
3 second local lock manager process executes.
- 1 18. (Original) A method as recited in Claim 16, wherein a first host on which the first
2 local lock manager process is executing is in a local network with a computing device
3 on which the client process is executing.
- 1 19. (Original) A method as recited in Claim 16, further comprising the steps of
2 determining whether a request from the client process to set the retry time period has
3 been received, and if so, automatically repeating the steps of determining, sending,
4 receiving and providing.

- 1 20. (Previously presented) A method of controlling concurrent users of a
2 distributed resource on a network, the distributed resource limited to a maximum
3 number of concurrent users, the method comprising the steps of:
4 receiving at a first local lock manager process of a distributed lock manager process a
5 request for a lock object for the distributed resource from a resource server,
6 wherein the request includes data indicating a particular user home location;
7 determining whether a second local lock manager process of the distributed lock
8 manager process is associated with the particular user home location, and if
9 so, requesting the lock object from the second local lock manager process,
10 wherein the first local lock manager process may grant a lock on the same resource as
11 the second local lock manager process.

1 21. (Original) A method as recited in Claim 20, further comprising the steps of:
2 when the second lock manager process is not associated with the particular home
3 location, determining whether a first number of outstanding locks granted by
4 the first local lock manager process for the distributed resource is less than a
5 first local maximum number of users no greater than the maximum number of
6 concurrent users, and if so, then incrementing the first number of outstanding
7 locks granted and providing the lock object to the resource server.

1 22. (Original) A method as recited in Claim 20, wherein each local lock manager
2 process of the distributed lock manager process stores a corresponding local
3 maximum number of users for the distributed resource in a lock data structure
4 associated with the local lock manager process, and an aggregate of local maximum
5 numbers in lock data structures associated with all lock manager processes in the
6 distributed lock manager process for the distributed resource is no greater than the
7 maximum number of concurrent users to which the distributed resource is limited.

1 23. (Original) A method as recited in Claim 20, further comprising the steps of:

2 receiving a request for a lock object from a third local lock manager process of the
3 distributed lock manager process in response to a request from a second
4 resource server;
5 determining whether a first number of outstanding locks granted by the first lock
6 manager process for the distributed resource is less than a first local maximum
7 number of users no greater than the maximum number of concurrent users;
8 and
9 if it is determined the first number of outstanding locks granted is less than the first
10 local maximum number, then
11 incrementing the first number of outstanding locks granted, and
12 providing the lock object for the second resource server.

- 1 24. (Original) A method as recited in Claim 20, further comprising the step of
2 providing the lock object to the resource server in response to receiving the lock
3 object from the second lock manager process.
- 1 25. (Previously presented) A method of distributing a resource on a network, the
2 resource limited to a maximum number of concurrent users, the method comprising
3 the steps of:
4 providing a distributed lock manager process comprising a plurality of local lock
5 manager processes executing on a corresponding plurality of hosts,
6 wherein each of the plurality of local lock manager processes may grant a lock on the
7 same resource;
8 generating a value for a local resource maximum number of users stored on each host
9 of the plurality of hosts such that a summation over the plurality of hosts of
10 the value for the local resource maximum yields an aggregate value that does
11 not exceed the maximum number of concurrent users;
12 determining whether to increase a first value in a first resource maximum stored on a
13 first host of the plurality of hosts; and
14 if it is determined to increase the first resource maximum, then

15 decreasing by a particular amount a second value in a second resource
16 maximum stored on a second host of the plurality of hosts, and
17 increasing by the particular amount the first value in the first resource
18 maximum stored on the first host,
19 wherein each local lock manager process is configured to grant a lock for the resource
20 if the number of outstanding locks granted by the local lock manager process
21 is less than a value of the local resource maximum stored on the
22 corresponding host.

- 1 26. (Original) A method as recited in Claim 25, said step of determining whether to
2 increase the first value further comprising determining the time of day and the
3 geographic location served by the first host.
- 1 27. (Original) A method as recited in Claim 25, said step of determining whether to
2 increase the first value further comprising determining whether a difference between
3 the number of outstanding locks granted and the first value is less than a first
4 predetermined threshold.
- 1 28. (Original) A method as recited in Claim 25, further comprising, if it is
2 determined to increase the first value, then determining whether to decrease the value
3 based on a whether a difference between the number of outstanding locks granted and
4 the second value is greater than a second predetermined threshold.
- 1 29. (Original) A method as recited in Claim 28, wherein the particular amount is
2 based on the second predetermined threshold.
- 1 30. (Original) A method as recited in Claim 29, wherein the particular amount is no
2 greater than the second predetermined threshold.
- 1 31. (Canceled)

- 1 32. (Currently Amended) A computer-readable storage medium carrying storing one or
2 more sequences of instructions for controlling a number of concurrent users of a
3 distributed resource on a network, the resource limited to a maximum number of
4 concurrent users, which instructions, when executed by one or more processors, cause
5 the one or more processors to carry out the steps of:
6 associating a user identification for each user with one host of a plurality of hosts on
7 which are executing a corresponding plurality of local lock manager processes
8 of a distributed lock manager process,
9 wherein each of the plurality of local lock manager processes may grant a lock on the
10 same resource at any time when a number of outstanding locks granted on the
11 same resource is less than the specified maximum number of concurrent users;
12 and
13 responding to a request for the resource associated with a first user having a first user
14 identification associated with a first host of the plurality of hosts by requesting
15 a lock from a first local lock manager executing on the first host.
- 1 33. (Currently Amended) A computer-readable storage medium carrying storing one or
2 more sequences of instructions for controlling a number of concurrent users of a
3 distributed resource on a network, the resource limited to a maximum number of
4 concurrent users, which instructions, when executed by one or more processors, cause
5 the one or more processors to carry out the steps of:
6 receiving a request for the distributed resource from a client process for a user having
7 a user identification;
8 determining one home location associated with the user identification, the home
9 location indicating a unique host among a plurality of hosts executing a
10 corresponding plurality of local lock manager processes of a distributed lock
11 manager process,
12 wherein each of the plurality of local lock manager processes may grant a lock on the
13 same resource; and

14 sending a request for a lock object for the distributed resource to a first local lock
15 manager process of the distributed lock manager process, the request
16 including the home location;
17 receiving the lock object for the distributed resource from a second local lock
18 manager process executing on the unique host, if a number of outstanding
19 locks granted by the second local lock manager process is less than a value of
20 a local resource maximum defined for the second local lock manager process;
21 and
22 providing access to the resource to the first client in response to receiving the lock
23 object.

1 34. (Currently Amended) A computer-readable storage medium carrying storing one or
2 more sequences of instructions for controlling a number of concurrent users of a
3 distributed resource on a network, the distributed resource limited to a maximum
4 number of concurrent users which instructions, when executed by one or more
5 processors, cause the one or more processors to carry out the steps of:
6 receiving at a first local lock manager process of a distributed lock manager process a
7 request for a lock object for the distributed resource from a resource server,
8 the request including data indicating a particular user home location, and the
9 first local lock manager process executing on a first host;
10 determining whether a second local lock manager process of the distributed lock
11 manager process is associated with the particular user home location, the
12 second local lock manager process executing on a second host different from
13 the first host,
14 wherein the first local lock manager process may grant a lock on the same resource as
15 the second local lock manager process; and
16 if it is determined that the second local lock manager process is associated with the
17 particular home location, then requesting the lock object from the second local
18 lock manager process.

- 1 35. (Currently Amended) A computer-readable storage medium carrying storing one or
2 more sequences of instructions for distributing a resource on a network, the resource
3 limited to a maximum number of concurrent users, which instructions, when executed
4 by one or more processors, cause the one or more processors to carry out the steps of:
5 generating a value for a local resource maximum number of users stored on each host
6 of a plurality of hosts executing a corresponding plurality of local lock
7 manager processes of a distributed lock manager process such that a
8 summation over the plurality of hosts of the value for the local resource
9 maximum yields an aggregate value that does not exceed the maximum
10 number of concurrent users,
11 wherein each of the plurality of local lock manager processes may grant a lock on the
12 same resource;
13 determining whether to increase a first value in a first resource maximum stored on a
14 first host of the plurality of hosts; and
15 if it is determined to increase the first resource maximum, then
16 decreasing by a particular amount a second value in a second resource
17 maximum stored on a second host of the plurality of hosts, and
18 increasing by the particular amount the first value in the first resource
19 maximum stored on the first host,
20 wherein each local lock manager process is configured to grant a lock for the resource
21 if the number of outstanding locks granted by the local lock manager process
22 is less than a value of the local resource maximum stored on the
23 corresponding host.

1 36. (Canceled)

1 37. (Previously Presented) An apparatus for controlling a number of concurrent
2 users of a distributed resource on a network, the resource limited to a maximum
3 number of concurrent users, comprising:
4 means for associating a user identification for each of the concurrent users with a
5 unique host; and

6 means for responding to a request for the resource associated with a first user among
7 the concurrent users who has a first user identification associated with a first
8 host by requesting a lock from a local lock manager process executing on the
9 first host.

10 wherein the local lock manager process may grant a lock on the same resource as
11 another local lock manager process at any time when a number of outstanding
12 locks granted on the same resource is less than the maximum number of
13 concurrent users.

1 38. (Previously presented) An apparatus for controlling a number of concurrent
2 users of a distributed resource on a network, the resource limited to a maximum
3 number of concurrent users, comprising:
4 means for receiving a request for the distributed resource from a client process for a
5 user having a user identification;

6 means for determining one home location associated with the user identification, the
7 home location indicating a unique host among a plurality of hosts executing a
8 corresponding plurality of local lock manager processes of a distributed lock
9 manager process,

10 wherein each of the plurality of local lock manager processes may grant a lock on the
11 same resource;

12 means for sending a request for a lock object for the distributed resource to a first
13 local lock manager process of the distributed lock manager process, the
14 request including the home location;

15 means for receiving the lock object for the distributed resource from a second local
16 lock manager process executing on the unique host, if a number of
17 outstanding locks granted by the second local lock manager process is less
18 than a value of a local resource maximum defined for the second local lock
19 manager process; and

20 means for providing access to the resource to the first client in response to receiving
21 the lock object.

- 1 39. (Previously presented) An apparatus for controlling a number of concurrent
2 users of a distributed resource on a network, the distributed resource limited to a
3 maximum number of concurrent users, comprising:
4 means for receiving at a first local lock manager process of a distributed lock
5 manager process a request for a lock object for the distributed resource from a
6 resource server, the request including data indicating a particular user home
7 location, and the first local lock manager process executing on a first host;
8 means for determining whether a second local lock manager of the distributed lock
9 manager is associated with the particular user home location, the second local
10 lock manager process executing on a second host different from the first host,
11 wherein the first local lock manager process may grant a lock on the same resource as
12 the second local lock manager process; and
13 means for requesting the lock object from the second local lock manager, if it is
14 determined that the second local lock manager is associated with the particular
15 home location.
- 1 40. (Original) An apparatus for distributing a resource on a network, the resource
2 limited to a maximum number of concurrent users, comprising:
3 means for generating a value for a local resource maximum number of users stored on
4 each host of a plurality of hosts in the network such that a summation over the
5 plurality of hosts of the value for the local resource maximum yields an
6 aggregate value that does not exceed the maximum number of concurrent
7 users;
8 means for determining whether to increase a first value in a first resource maximum
9 stored on a first host of the plurality of hosts; and
10 means for decreasing by a particular amount a second value in a second resource
11 maximum stored on a second host of the plurality of hosts, if it is determined
12 to increase the first resource maximum;

13 means for increasing by the particular amount the first value in the first resource
14 maximum stored on the first host, if it is determined to increase the first
15 resource maximum;
16 means for granting a lock for the resource if the number of outstanding locks granted
17 by the local lock manager is less than a value of the local resource maximum
18 stored on the corresponding host.

1 41. (Canceled)

1 42. (Canceled)

1 43. (Canceled)

1 44. (Canceled)

1 45. (Previously presented) An apparatus for controlling use by concurrent users of
2 a distributed resource on a network, wherein use of the resource is limited to a
3 specified maximum number of concurrent users, comprising:
4 a host that is communicatively coupled to a network that contains the distributed
5 resource and that comprises one or more processors and a computer-readable
6 storage medium;
7 one or more sequences of instructions stored in the computer-readable storage
8 medium which, when executed by the one or more processors, cause the one
9 or more processors to carry out the steps of:
10 receiving, at a first local lock manager process of a distributed lock manager
11 process, a request for a lock object for the distributed resource from a
12 resource server, wherein the request includes data indicating a
13 particular user home location;
14 determining whether a second local lock manager process of the distributed
15 lock manager process is associated with the particular user home
16 location, and if so, requesting the lock object from the second local
17 lock manager process,

18 wherein the first local lock manager process may grant a lock on the same
19 resource as the second local lock manager process.

1 46. (Previously presented) An apparatus as recited in Claim 45, wherein execution
2 of the one or more instructions by the one or more processors further causes:
3 when the second lock manager process is not associated with the particular home
4 location, determining whether a first number of outstanding locks granted by
5 the first local lock manager process for the distributed resource is less than a
6 first local maximum number of users no greater than the maximum number of
7 concurrent users, and if so, then incrementing the first number of outstanding
8 locks granted and providing the lock object to the resource server.

1 47. (Previously presented) An apparatus as recited in Claim 45, wherein each local
2 lock manager process of the distributed lock manager process stores a corresponding
3 local maximum number of users for the distributed resource in a lock data structure
4 associated with the local lock manager process, and an aggregate of local maximum
5 numbers in lock data structures associated with all lock manager processes in the
6 distributed lock manager process for the distributed resource is no greater than the
7 maximum number of concurrent users to which the distributed resource is limited.

1 48. (Previously presented) An apparatus as recited in Claim 45, wherein execution
2 of the one or more instructions by the one or more processors further causes:
3 receiving a request for a lock object from a third local lock manager process of the
4 distributed lock manager process in response to a request from a second
5 resource server;
6 determining whether a first number of outstanding locks granted by the first lock
7 manager process for the distributed resource is less than a first local maximum
8 number of users no greater than the maximum number of concurrent users;
9 and
10 if it is determined the first number of outstanding locks granted is less than the first
11 local maximum number, then
12 incrementing the first number of outstanding locks granted, and

13 providing the lock object for the second resource server.

- 1 49. (Previously presented) An apparatus as recited in Claim 45, wherein execution
2 of the one or more instructions by the one or more processors further causes:
3 providing the lock object to the resource server in response to receiving the lock
4 object from the second lock manager process.